事件属性抽取任务

1. 任务概述

已知一批自然语言文本数据，每个数据样本为一个句子，如下所示。句子内部包含若干“事件”，我们通常标注出“触发词”来表示事件的存在，每个句子当中我们会标注出两个触发词。例如，下列句子当中 tossed 和 choked 均为事件的触发词。

Police said two women and one man , between 30 and 40 years old , working for a Marfin bank branch <e2> choked </e2> on smoke after protesters broke the windows of a commercial building on central Stadiou Avenue and <e1> tossed </e1> in Molotov cocktails .

“触发词”一般为动词和名词，但是少数情况下也可以是形容词和副词。现在，我们希望从触发词开始，逐步提取事件的全部属性：事件主体/事件主体修饰 、 事件动作/事件动作修饰 、 事件客体/事件客体修饰。
例如，在上述例子当中，两个事件分别为:

- [two women and one man, between 30 and 40 years old], [choked, on smoke], []
- [protesters,], [tossed,], [Molotov cocktails,]

2. 提取规则

2.1 事件提取要求

事件提取需遵循下述规则，事件的主体、动作、客体都可以缺省，但是至少要有主体和动作其中之一，触发词必须包含在事件属性当中。各个部分的修饰都可以缺省，找到的越多越好，不需要标注出修饰的类型，各部分提取出来的单词直接沿用原文即可。如果遇到事件属性是一句话的情况，可以使用关键词来代替。

事件属性	描述
主体	事件的发起者，一般为人，集体，物品，如： Women、America。 也可以是一个名词或者合成名词，如：Earthquake、Word War-II。
主体修饰	事件主体具备的某些属性，如：年龄、强度等级、好坏等等。
动作	事件的主要变化，主要发展趋势，一般为动词，如： Choke
动作修饰	对事件发展趋势方向、程度等的修饰，一般为副词，如：Hard 也可以是介宾短语，表示原因或者事件时间地点，如：On somke，In hispital
客体	受到事件动作影响人，集体，物品，如： Moltotov cocktails
客体修饰	受到影响的程度、方向，如：Popular，Dead

在最后的评估指标当中，事件骨架的重要程度 > 事件修饰的重要程度。

2.2 提取结果数据格式

结果保存为 jsonl 格式，每一行为一个 json 对象，一个样本对应两行（两个事件）。事件格式如下：

```
JSON
{
   "subj": ".....",
   "mov": ".....",
   "obj": ".....",
   "subj_mod": ["....", "....", "...."],
   "mov_mod": ["....", "....", "...."],
   "obj_mod": ["..."]
}
```

2.3 推荐研究方法

本节介绍的研究思路仅供参考。由于事件 schema 结构和句子语法结构相似，因此采用依存句法分析进行研究，常用的依存句法分析工具有：spacy 和 stanford-nlp tool。下面代码展示了提取事件骨架的大体思路。由于在依存树当中，谓语常常位于树的根部，较为容易定位，因此先寻找事件的动作，然后基于此寻找事件的主体和客体。获取到事件骨架之后，即可继续去寻找各个属性的修饰。

Python

```
def extract(trigger):
    """
    Extract event and event attribute vai mention tokens generate by spacy
    Args:
        trigger: event trigger words

    Returns:

    """
    movement = find_movement(trigger)
    subject, passive = find_subject(movement, trigger, exclude=[movement])

    exclude = [movement]
    if isinstance(subject, list):
        exclude.extend(subject)
    else:
        exclude.append(subject)

    obj = find_object(movement, trigger, exclude=exclude)
    if passive or subject == obj:
        obj = None

    return (subject, passive), movement, obj
```